ITCE341: Lab 3 – 8086 Emulator and Arithmetic Instruction

Done by: Abeer(www.UOB-BH.com)

Objective

The objective of this experiment is to:

- **Familiarize with the EMU8086 Emulator.**
- **Implement Variables and some Arithmetic & Logic Instructions.**

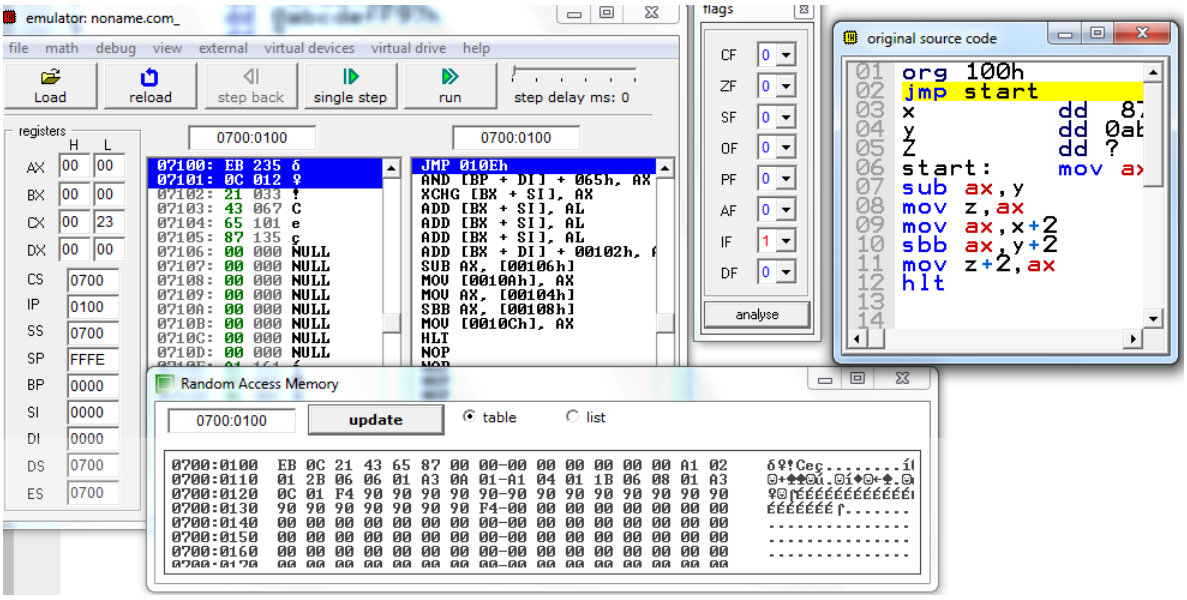
Procedure

Run emu8086 program

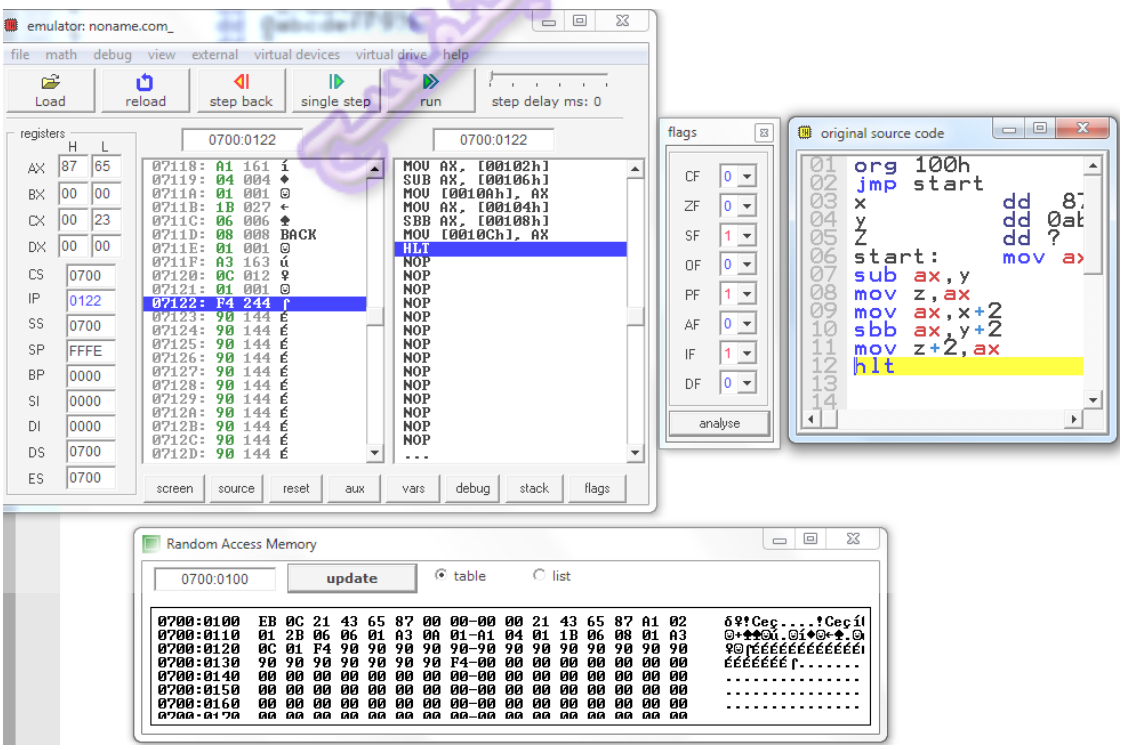
1. **In the welcome window click new button**
2. **In the choose template window select empty workspace**
3. **In the Source Code Editor window type the program as shown in next page**

```
        Org100h
        Jmp start

X       dd 87654321h

Y       dd 0abcdef97h

Z       dd ?

Start:   mov  ax,x
        Sub ax,y
        Mov z,ax
        Sbb ax,y+2
        Mov z+2,ax
        hlt
```

4. **Simulate the program by clicking on emulate button.**
5. **To view the effect on the processor flags during execution, open the flag window by clicking on the flag button, the flag window will appear.**
6. **To view the effect on memory during execution, open the memory window by clicking on the aux button then memory, the memory window will appear**
7. **Move the flags and memory windows to proper locations so that you can view them simultaneously with the main emulator window**
8. **View and record the contents of all registers, flags and memory**

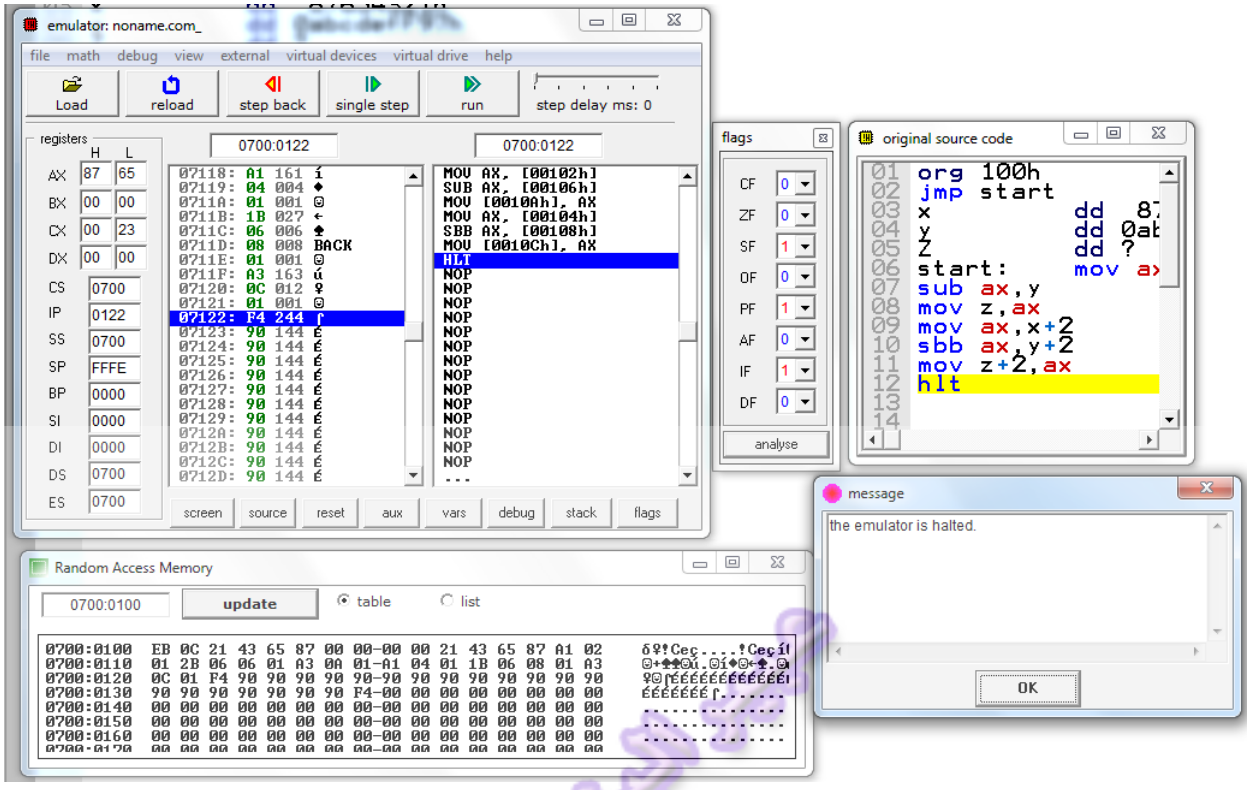


9. **Press on the step button to execute the program step by step , and view carefully the affected registers, flags and memory. Record them all**



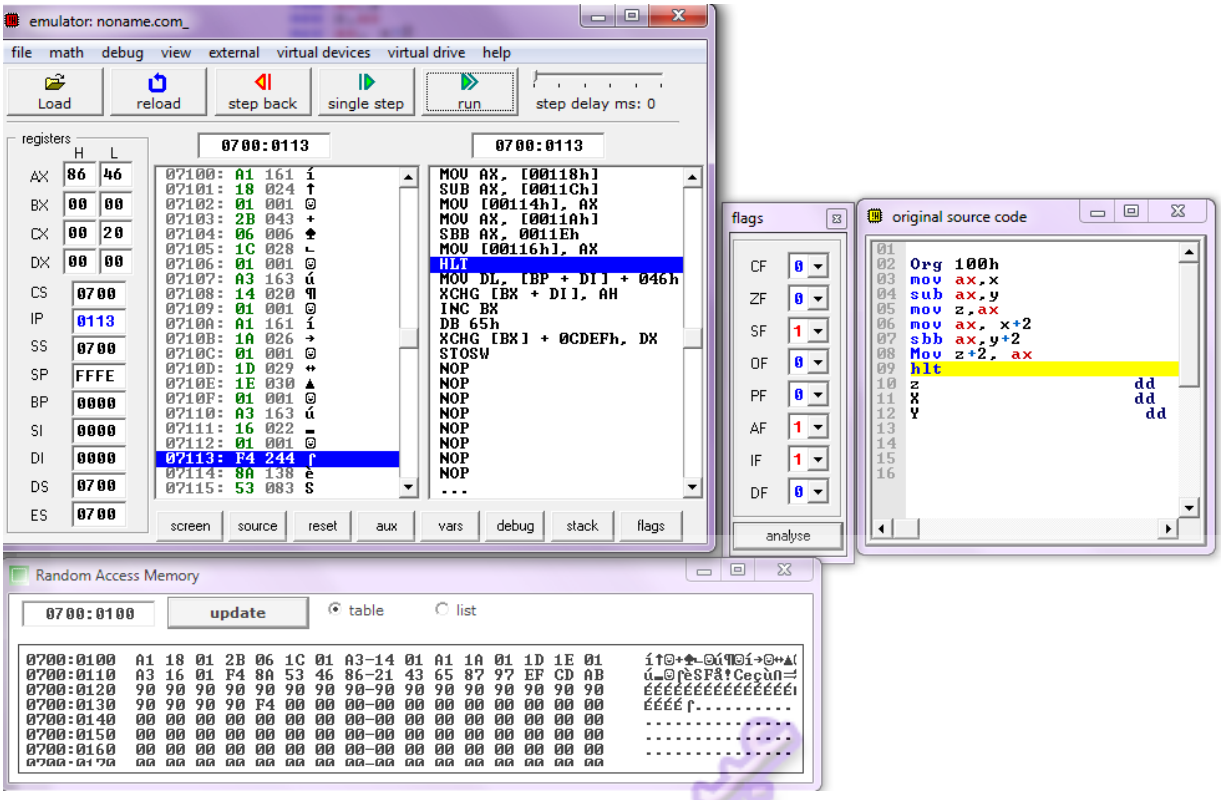
10. **Reload the program and run**

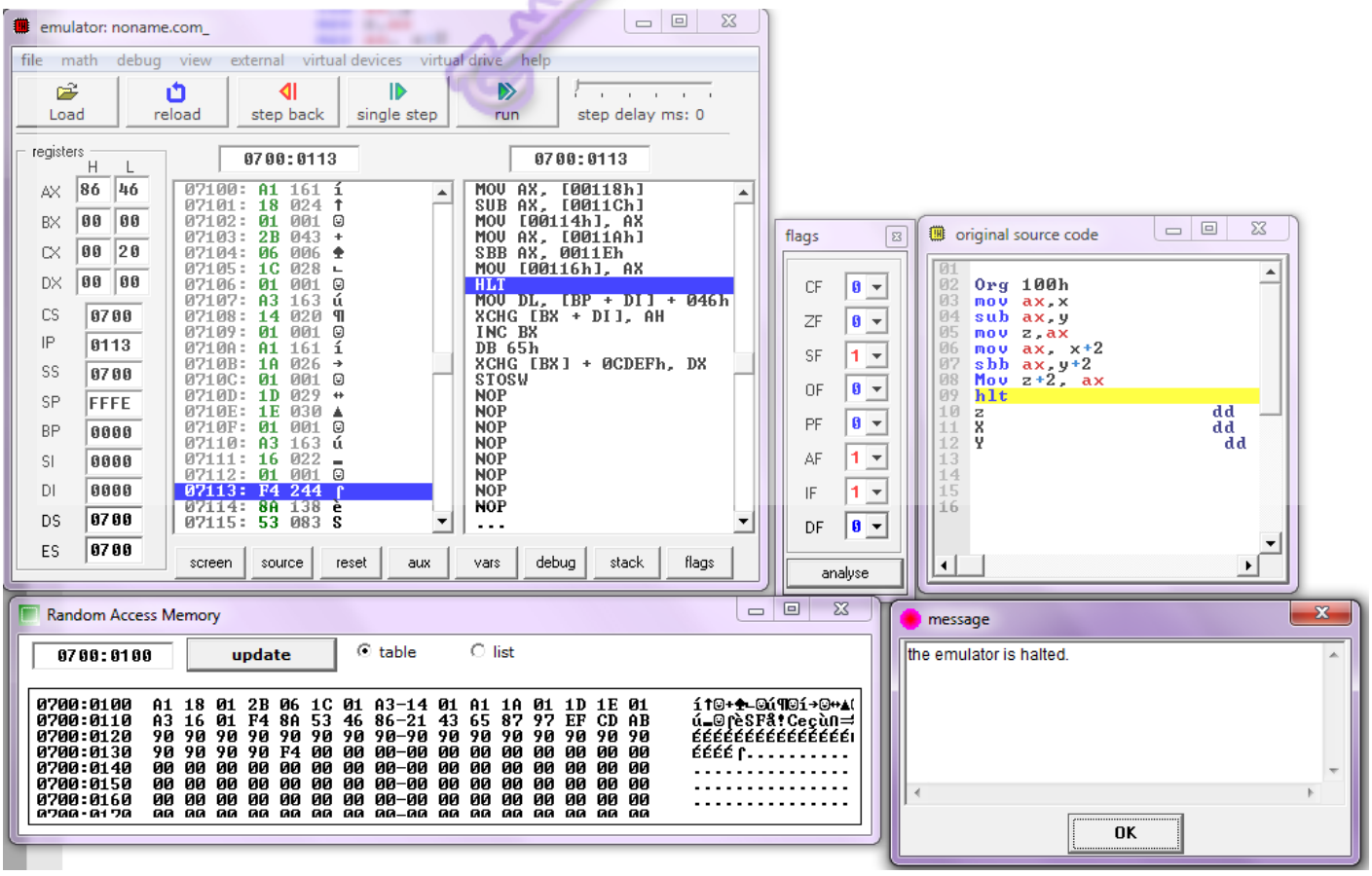
Done by: Abeer(www.UOB-BH.com)



11. Repeat steps 3 to 10 with the following modified version of the same program:

```
                Org 100h
                mov ax,x
                sub ax,y
                mov z,ax
                mov ax, x+2
                sbb ax,y+2
                Mov z+2, ax
                hlt
z               dd  ?
X               dd  87654321h
Y               dd  0abcdef97h
```

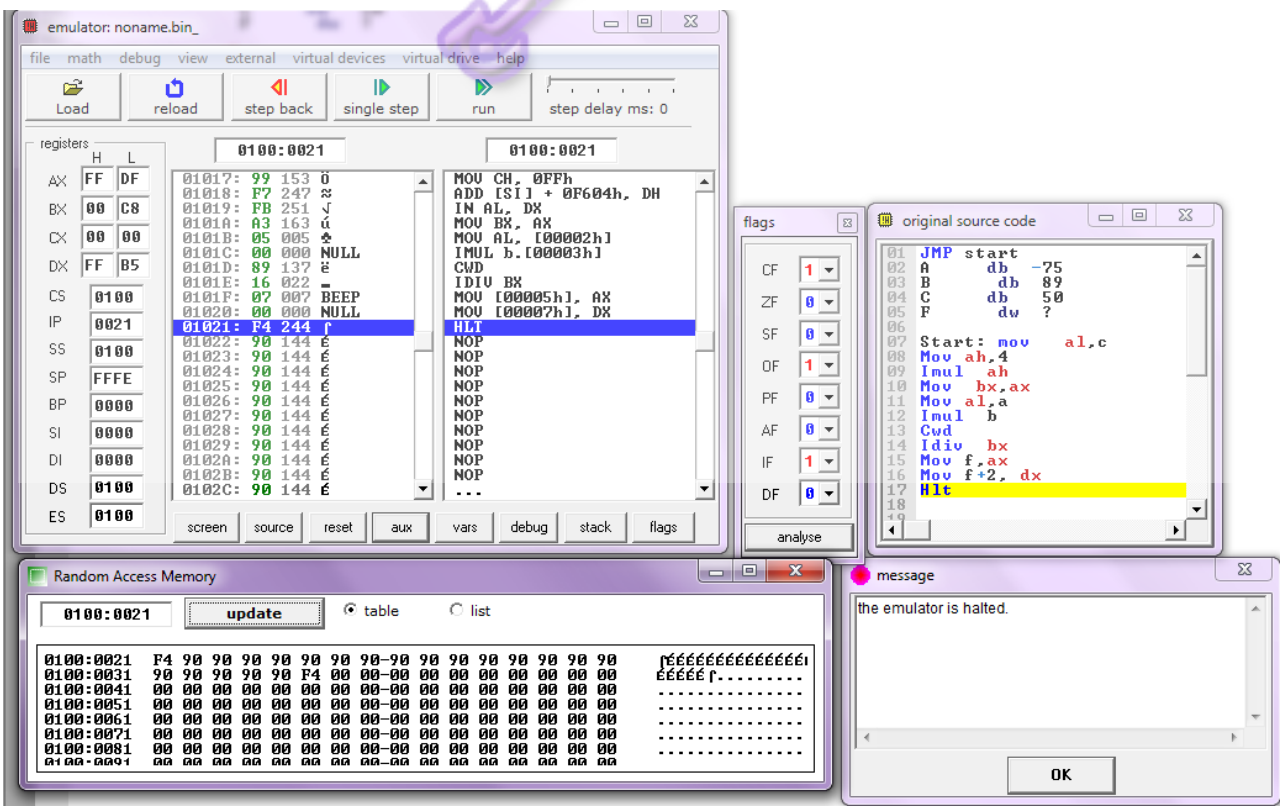




12. Write a program (terminated with hlt instruction) to perform the following operation, where a,b and c are signed bytes:
f= 0.25* a* b÷c

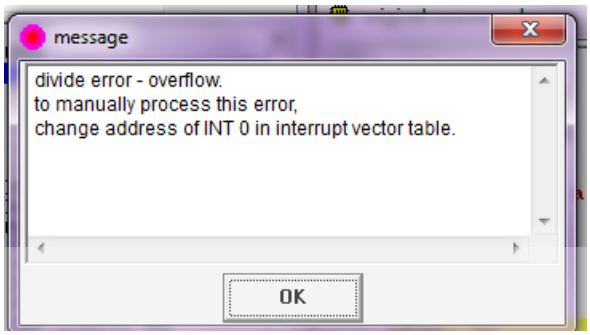
```
JMP start
A    db  33
B    db  55
C    db  66
F    dw  ?

Start: mov   al,c
       Mov ah,4
       Imul  ah
       Mov  bx,ax
       Mov al,a
       Imul  b
       Cwd
       Idiv  bx
      Mov f,ax
      Mov f+2, dx
     Hlt
```

13. Implement the program with a=-75, b=89 & c=50 execute the program step by step with single step. Reload it and run it with run.



14. Repeat step 13 with a=375, b=89, c=00.

It said there is an error



Report:

1. **What would happen if the instruction jmp strt were not there?**

   It will execute the defining variables and it will take them as instructions and We will start from data, and this is wrong because we must start from an instruction (jmp strt).

2. **Why the instruction jmp strt is not needed in the second version?**

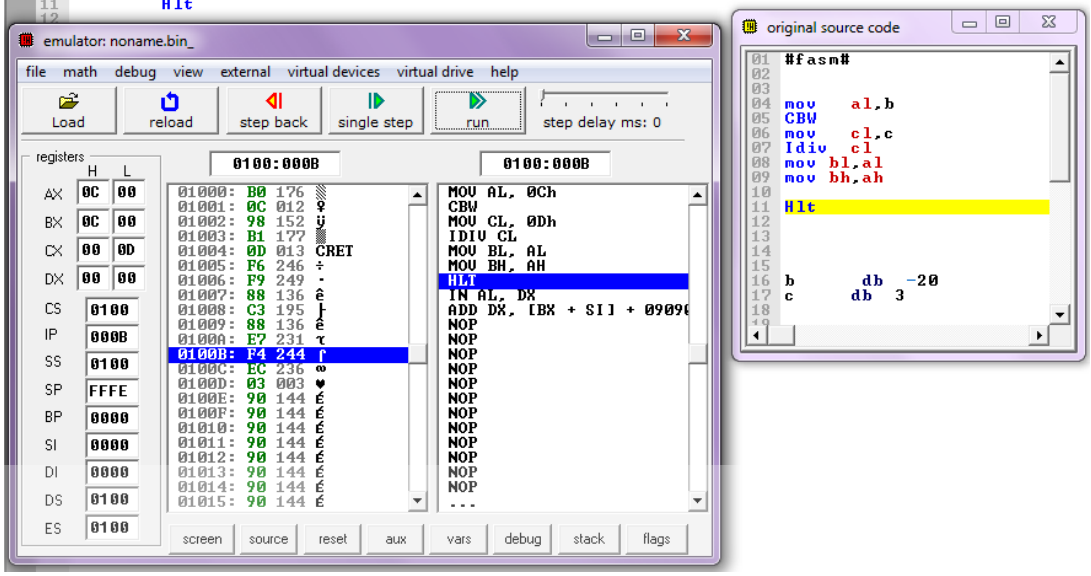
   Because the definition is below the program after halting it, so it will not take it as part of the program (will not take as instructions)

3. **Discuss what differences did you notice between program flow execution and results in step 13 with that of step 14.**

   There is one different when we execute the program in step 14 which is an error in the division (overflow) because if the 00

4. **What would be the quotient and the remainder when you divide (-20) by (3) ?**
   Realize youe answer with 86-emulator.

```
mov    al,b
       CBW
       mov    cl,c
       Idiv   cl
       mov bl,al
       mov bh,ah

        Hlt
```



The quotient is 0 and the remainder is 0C

5. What would be the the quotient and the remainder when you divide (20) by (-3) ?
   Realize youe answer with 86-emulator.

The quotient is 0 and the remainder is 0C

```
mov    al,b
       CBW
       mov    cl,c
       Idiv   cl
       mov bl,al
       mov bh,ah

        Hlt
```